

國立中山大學 111 學年度

資訊工程學系專題計劃書

**應用神經網路架構搜索技術提升 YouBike 系統之
車流預測準確度**

**An Effective Evolutionary Neural Architecture
Search for Bike-Sharing System Demand
Prediction**

組員：

B084020005 蔡昀燁

B083040012 陳柏翰

B083040022 黃兆延

指導老師：

蔡崇煒 老師

1 摘要

近年隨著環保意識和智慧城市的影響，公共自行車系統 (bike-sharing system ; BSS) 因為其環保和便利的特性，已然成為各大城市重點發展的公共設施。因各站點站位有限，可能產生使用者鄰近站點車輛數不足和無站點可停的問題，若工作人員無法即時檢修與調度而造成站點的車輛數和空位不平衡，將會降低使用者使用意願。本專案提出超啟發式演算法 (adaptive simulated annealing genetic algorithm ; ASAGA) 作為 one-shot 神經網路架構搜尋 (neural architecture search ; NAS) 的搜尋策略，搭配上以混合為基礎的超參數 (hyperparameter) 優化，藉由紐約市公共自行車系統資料，建立高效深度學習模型用以預測各站點狀況，並實際應用於本地高雄市公共自行車系統以解決各站點車輛不平衡之問題。期望以預測結果改善現有公共自行車的服務品質，提高民眾使用意願，最終達到紓困交通且提升營運方收入的效果。

關鍵詞: 神經網路搜索、超參數優化、自行車使用量預測

目錄

1	摘要	1
2	研究動機與研究問題	4
3	文獻回顧與探討	6
3.1	交通流預測模型相關研究	6
3.2	腳踏車共享系統之站點預測相關研究	7
3.3	神經網路架構搜尋相關研究	8
3.4	One-Shot NAS 及搜尋策略相關研究	9
4	研究方法	11
4.1	空間與時間的相互依賴性	12
4.2	One-Shot NAS 及自適應退火基因演算法	13
4.3	興趣指數	17
4.4	最佳化模型超參數	18
5	實驗	20
5.1	環境設置	20
5.2	神經網路架構搜尋前處理	21
5.2.1	再平衡對於公共自行車系統的影響	21
5.2.2	車流量	21
5.2.3	站點車輛數目	22
5.2.4	參數設定	22

5.3	超參數優化前處理	23
5.3.1	編碼設計	23
5.3.2	參數設定	23
5.4	實驗結果	24
5.4.1	神經網路架構搜尋成果	24
5.4.2	超參數優化成果	26
5.5	使用者介面設計	26
6	結論	30
7	評審意見與回覆	31
8	致謝和聲明	33
9	參考文獻	34

2 研究動機與研究問題

公共自行車系統已被許多城市使用多年，隨著民眾對自行車系統的認識，其便利性、環保等優勢，讓使用人數大幅提升。使用量的增加產生固定站點的公共自行車系統 (dock bike-sharing system) 站點車輛數不平衡的問題 [1]，也就是無車可借以及無位可還的狀況，此問題影響到使用者的體驗。各站點的調度問題牽涉到的因素相當複雜，站點自行車數量的變化大致上受到站點周遭人、車流以及使用者習慣兩個重要因素影響；站點附近的人、車流會因為站點位置、城市中的大眾交通運輸結構和地理環境而有所不同，民眾的使用習慣則會受到當下天氣、道路交通狀況和騎乘用途的影響。由於公共自行車有著騎乘路線不固定以及不限於在同一站點租借、歸還的特性，公共自行車系統的站點需求預測至今仍是各廠商致力完成的目標，許多學者也開始將日漸發展成熟的深度學習技術應用於公共自行車的實務問題上，嘗試解決這道複雜的難題。

近年來由於深度學習在電腦視覺領域獲得突破性的發展 [2]，將深度學習應用於自行車站點車輛數預測成為本專案的研究重點。目前所被大眾熟知的深度學習模型包括 VGG16 [3]、ResNets [4] 等皆為卷積神經網路 (convolutional neural network ; CNN)，其處理的資料多為圖像，然而不同於許多影像辨識的資料集，處理公共自行車系統資料時不只需要考慮各站點之間所蘊含的空間特徵，其時序性也是重要的參考依據，因此空間-時間混合的神經網路模型 (spatial - temporal hybrid model) 成為現今預測交通流的重要方法之一。

為了讓深度學習模型在其應用的問題上得出最佳的結果，必須對模型的架構和參數進行調整，包括資料的選擇、超參數的數值 (hyperparameter tuning) [5]、資料的前處理 (data preprocessing) [6] [7] 以及神經網路架構的設計。進行人為調整需要特定領域的專業背景知識並花費大量時間，能夠自動生成高預測準確率模型架構的自動化機器學習 (automated machine learning) [8] 因而成為近年的熱門主題，其中又以神經網路架構搜尋 [9] 最受矚目。然而在大量的超參數需要被考慮進 NAS 的搜尋空間 (search space) 時，使用 NAS 建立合適的模型將需要大量的計算資源 [10]。為了解決此問題，本專題製作利用 one-shot NAS 方法 [11] 建立一個超級網路 (supernet)，並提出 ASAGA 作為搜尋策略，生成最佳的神經網路架構預測腳踏車系統中各站點的車輛數，減少搜索時間與大量的計算資源消耗，期望在有限時間、計算資源的情況下獲得高準確度的結果，並投入至本地公共自行車系統來解決腳踏車分配不平衡的問題。

3 文獻回顧與探討

3.1 交通流預測模型相關研究

交通流 (traffic flow) 擁有高度的時序及空間之關聯性質，卷積長短期混合模型 (CNN-LSTM hybrid model) 因而成為眾多研究所使用的預測方法。利用卷積神經網路提取空間特徵及長短期記憶模型 (long short-term memory ; LSTM) 挖掘時間特徵，把交通流的時間及空間特性一同進行處理。Narmadha 等人 [12] 建構了多變量的卷積長短期記憶混合模型 (multivariate CNN-LSTM)，藉由混合一維卷積層 (CNN layers) 及長短期記憶的框架來預測多變量的輸入，並與其他五種監督式學習方法比較，實驗結果顯示混合模型的準確度超越了其他的非混合模型。Zheng 等人 [13] 則是根據交通流所蘊含的周期性，推斷不同日期的同個時段可能會有相似的車流與人流變化，基於此理論在基礎的卷積長短期記憶混合模型上搭配雙向長短期記憶模型 (Bi-directional LSTM)，得到更為完整的時間性資料，並從實驗結果證實一定程度上參照未來的時間段能夠有效提升預測準確率。Mihaita 等人 [14] 對混合模型提出了疑問並實驗測試，將卷積神經網路及長短期記憶模型分離出來，發現單獨使用長短期記憶模型比起卷積長短期記憶混合模型更加準確，推斷卷積神經網路在某些狀況下反而會拖累整體表現。

3.2 腳踏車共享系統之站點預測相關研究

現今的腳踏車共享系統大多具有固定站點的特性，在與系統相關的應用問題中，經常使用站點的車輛變化作為預測目標，例如：調整各站點自行車數量來改善使用者體驗、分析各站點資訊來決定是否需要增設站點 [15] 等。Lu [16] 及 Zhou [17] 等人透過在資料集中加入興趣指數 (point of interest ; POI)，突顯各站點的空間特徵，並利用循環神經網路 (recurrent neural network ; RNN) 進行訓練，有效地提高了預測站點車輛變化的準確度。Yang 等人 [18] 以圖論為基礎，針對各站點新增特徵點，例如單一站點在特定時間內的車輛流入量 (inflow) 等，並利用機器學習模型預測紐約市共享自行車的各站點需求量。

不少研究都開始採用深度學習的方式 [19] 來進行交通流的預測。為了取得用傳統機器學習方法無法發覺的潛藏特徵，越來越多的研究將交通資料當作類影像資料並使用卷積神經網路來捕捉其中的空間資訊。Shi [20] 等人建立了卷積長短期記憶模型 (convolutional long short-term memory) 來處理空間與時間的相互依賴性 (spatio-temporal dependency)，在降雨預測上獲得不錯的實驗成果，證明了時間與空間的關聯性 (spatio-temporal correlation) 對於預測連續時序性的目標是非常關鍵的。Yao 等人 [21] 提出了空間-時間混合的動態深度學習框架 (spatial-temporal dynamic network ; STDN) 來進行交通流預測，不像是先前的研究將空間與時間的相互依賴性離散化地處理，STDN 藉由同時考慮每個站點之間的空間資訊與時間連續關係，妥善地處

理交通流資料空間與時間的相互依賴性，實驗結果表明了 STDN 比起其他方法有更高的準確率。

3.3 神經網路架構搜尋相關研究

神經網路架構搜尋的目的為利用一套演算法或是框架自動根據需求找出最好的神經網路架構，最早的神經網路架構搜尋由 Zoph 等人 [22] 所提出，基於強化學習 (reinforcement learning)，利用控制器 (controller) 從搜尋空間中找出一個神經網路架構進行訓練，並根據評價 (evaluate) 結果所獲得的獎勵 (reward) 來更新控制器，讓控制器學習如何從搜尋空間中找出更好的神經網路架構。但在沒有限制搜尋空間的情況下，利用強化學習作為搜尋策略需要反覆多次訓練模型，造成搜尋成本相當龐大，如果沒有足夠的運算資源會非常耗費時間，無法應用在真實世界的問題上。Wang 等人 [23] 有鑑於神經網路架構搜尋花費時間過長，改良並採用粒子群最佳化 (particle swarm optimization ; PSO) 來優化整體框架。藉由引入網路協定地址 (internet protocol address ; IP address) 的編碼方式及殘缺層 (disabled layer)，解決傳統粒子群演算法無法應用於可變層數 (variable-length) 的問題，實驗結果也證明可變層數的神經網路架構搜尋在準確率上有顯著提升。Sun 等人 [24] 提出了演化式卷積神經網路 (evolutionary convolutional neural network ; EvoCNN)，以基因演算法 (genetic algorithm ; GA) 為基礎，使用標準差 (standard derivation) 及平均值 (mean value) 為代表性資料，減少所需的運算資源與時間，結果表明演化式卷積神經網路能夠以更少的參數 (parameters) 在更短

的的訓練週期 (epoch) 得到更好的結果。

3.4 One-Shot NAS 及搜尋策略相關研究

Brock 等人 [25] 訓練一個超級網路來生成不同模型架構的權重，利用超級網路和常規訓練兩者分別所得權重之模型在預測結果上的高度相關性，驗證大量的神經網路架構並依照性能進行排序，更有效率地篩選出性能較佳的模型架構。首先建立一包含所有候選模型架構 (candidate architecture) 的超級網路，在每一次的訓練過程中隨機取樣一網路架構並利用超級網路生成該架構的權重，接著再利用所得誤差對超級網路進行反向訓練。超級網路訓練完畢後，從中隨機取樣架構並使用先前所得權重驗證其性能，藉以篩選出性能較佳的模型，這樣的方法透過訓練一次超級網路而非重複訓練不同架構來減少搜尋的時間。根據不同訓練超級網路的方法，許多 one-shot NAS 框架的變體也相繼被提出。高效率神經網路架構搜尋 (efficient neural architecture search ; ENAS) 使用近似梯度 (approximate gradients) [26] 來進行強化學習，模擬結果顯示 ENAS 在 CIFAR-10 資料集只有 2.89% 的錯誤率。可微分式神經網路架構搜尋 (differentiable architecture search ; DARTS) [27] 以梯度優化為主軸，實驗結果顯示 DARTS 在影像分類及語言模型化的任務準確率上超過了當代不使用微分策略架構搜尋之框架，並且有更好的搜尋效率。

單路徑 one-shot 神經網路架構搜尋 (single path one-shot neural network architecture search : SPOS) [28] 在訓練階段採用均勻抽樣 (uniform sampling) 來讓所有子架構獲得平等的訓練，並使用演化式演算

法 (evolutionary algorithm) 評估架構，結果顯示 SPOS 在 ImageNet 資料集上可以有效的提升準確率。在神經網路架構搜尋的框架中，搜尋策略對於搜尋的成果和時間成本有著決定性的影響，許多研究開始使用演化式演算法來加速模型的評量 [29]；Real [30] 的研究顯示相對於隨機搜尋和強化式學習，使用演化式演算法來搜尋架構可以取得更好的效率及準確率。

4 研究方法

Table 1: 符號表

符號	定義
a	興趣點地圖的種類 (在此研究中有 10 種分類)
i	站點 i
x	興趣點地圖的座標
$M_{i,x}^a$	在種類 a 、站點 i 、座標 x 網格中的興趣指數
$\mu_{i,x}^a$	在種類 a 、站點 i 、座標 x 網格中的評價總數
γ	在種類 a 、站點 i 、座標 x 網格中興趣地點的數量
ρ	在單個網格中的平均評價

為了能讓模型更好地理解交通流資料中空間與時間的相互依賴性以及利用神經網路架構搜尋建立更好的架構，我們提出了一個全新的神經網路架構搜尋框架-模擬退火基因演算法 one-shot 神經網路搜尋 (simulated annealing-genetic algorithm one-shot network ; SAGAON)。Fig. 1 為 SAGAON 框架的示意圖，主要分為 3 個部分。首先 Fig. 1(A) 中的複數卷積層捕捉空間特徵，其結果會與 Fig. 1(B) 的額外特徵進行串接。在得到空間特徵之後，資料將會被送到 Fig. 1(C) 部分的長短期記憶模型來解析時序特徵。最後 SAGAON 將會得到一個超級網路並使用 ASAGA 演算法來搜尋與公共自行車需求問題相對應的最好架構。

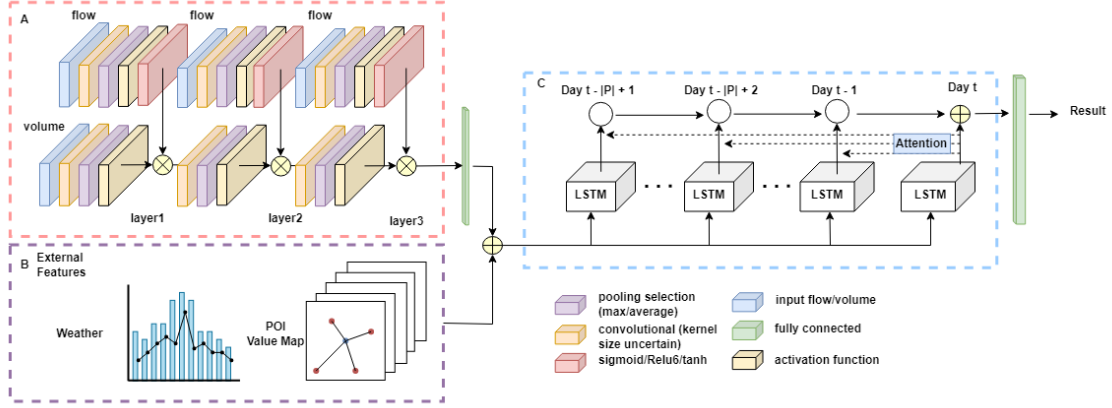


Fig. 1: SAGAON 的架構。(A) 站點車輛數目跟車流量資料會經過數層卷積層。one-shot NAS 會同時訓練多個包含不同卷積核大小、池化層和激勵函數的選擇模塊。(B) 採用天氣和興趣指數等額外特徵，該模型可以學習更多影響到公共自行車系統行為模式的外部因素。(C) 使用長短期記憶模型來捕捉腳踏車車流的週期性。 P 代表長短期記憶模型所考慮的天數，而 t 則是代表預測目標的時段 (timeslot)。

4.1 空間與時間的相互依賴性

Li 等人 [31] 根據地理位置將站點分群，並以群為單位進行預測，藉此減少模型的複雜度，不過考慮到公共自行車系統的真实使用情境，我們選擇預測單一站點的車流，並利用分群結果計算車流量 (flow) 特徵來增強每個站點對彼此流量關係的理解。我們使用 3 層的卷積層來讀取空間特徵，並以每個站點的站點車輛數目 (volume) 以及車流量為輸入。在每一層中，這兩種輸入都會經過 3 個線性選擇模塊 (linear choice block)，分別是 (1) 卷積核大小 (kernel size) (2) 最大或平均池化層 (max/average pooling) (3) 激勵函數 (activation function)，

在時段 t 第 i 站點的卷積層結果表示成 $R_{i,t}$ 。Fig. 1(C) 部分採用跟 STDN 相似的結構，由複數的長短期記憶層 (LSTM layer) 組成，為了將不同時間尺度的腳踏車車流週期性納入考量，每個長短期記憶層都會接收 Fig. 1(A) 前幾個時間段的輸出並加上 Fig. 1(B) 的額外特徵 (天氣和興趣指數) 來獲取空間及時間特徵。在這裡我們使用注意力機制 (attention mechanism) 來評斷每個時間段對時序預測的影響，用以捕捉近期車流週期性的細微差異。我們使用的長短期記憶模型是 Hochreiter [32] 所提出的原始版本， $R_{i,t}$ 和 $e_{i,t}$ 分別為站點 i 在時間段 t 的卷積結果與額外特徵， $h_{i,t}$ 則代表長短期記憶模型的輸出結果。長短期記憶模型的公式如下：

$$h_{i,t} = \text{LSTM}([R_{i,t}; e_{i,t}], h_{i,t-1}). \quad (1)$$

4.2 One-Shot NAS 及自適應退火基因演算法

為了減少搜尋架構的時間，Guo [28] 提出了採用均勻抽樣的單路徑 one-shot 神經網路架構搜尋。透過這個方法能夠減少搜尋時間，同時維持搜尋成果的準確率。Fig. 2 展示了元件 (cell component) 的內部狀況，基於 STDN 的原有架構，我們在每個超級網路的元件裡面建立了 3 個選擇模塊，將不同的卷積核大小 (kernel size) 和激勵函數 (activation function) 加入 SAGAON 的搜尋空間，並新增池化層 (pooling layer) 擴充原有的架構，除了上述的選擇外，移除特定的架構也是搜尋空間中的一種選擇，藉此增加可搜尋的模型架構數量。在 SAGAON 裡，三種不同層級的元件被設置在處理空間特徵

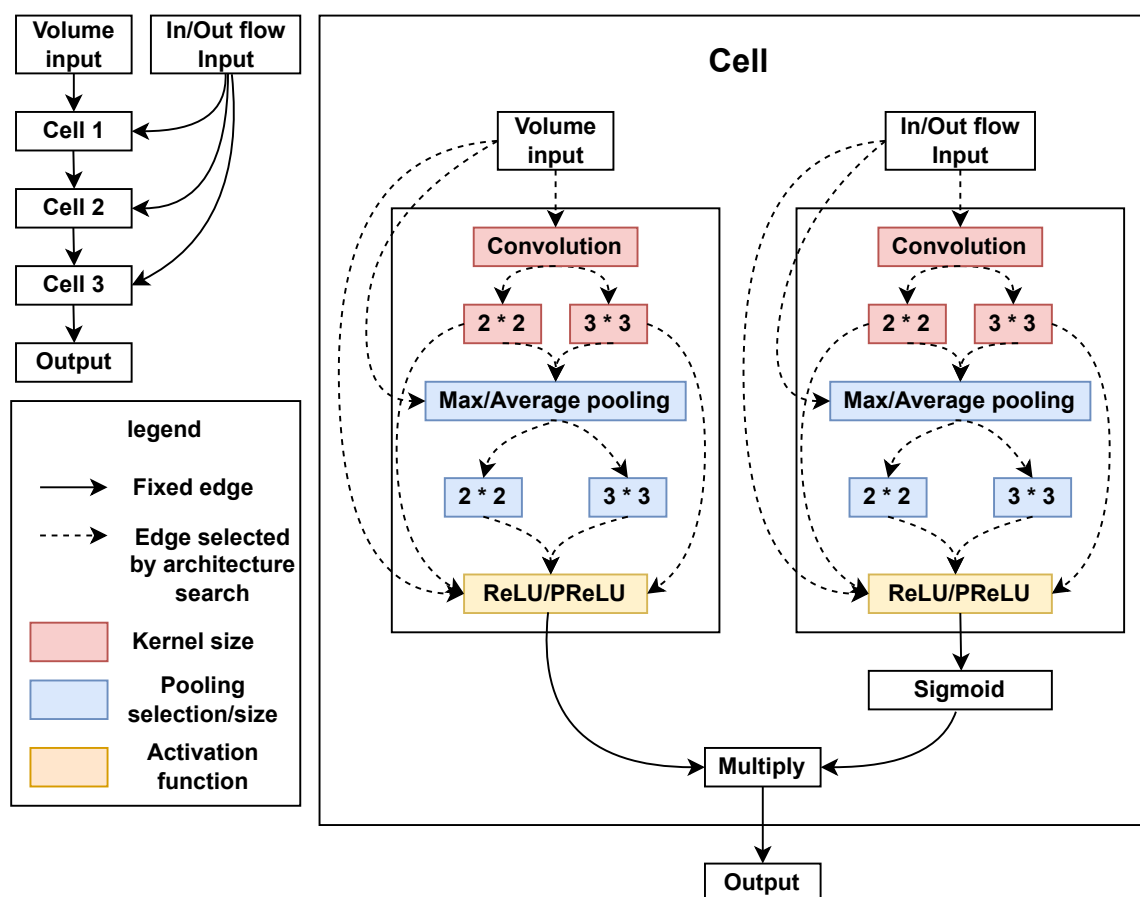


Fig. 2: 元件與選擇模塊。站點車輛數特徵以及車流量特徵會經過三層元件，並在選擇模塊中隨機選擇不同的卷積核大小、池化層以及激勵函數。圖中的實線為固有的模型架構，虛線為經過隨機選擇的可能結果。

的 Fig. 1(A) 部分。為了避免不同大小的矩陣發生無法進行計算的情況，經過卷積和池化後將會使用零填充 (zero padding) 維持矩陣大小的一致性，保證接下來的矩陣乘法可以順利運行。車流量輸入的部分，在進行矩陣乘法之前會先經過 S 型函數 (sigmoid function)，避免車流量的特徵值成長過於迅速。在產生超級網路之後，我們使用了結合基因演算法和模擬退火演算法的 ASAGA 搜尋最佳神經網路架構，虛擬碼如 Algorithm 1 所示，交配操作子 (crossover operator) 藉由隨機交換兩個親代的片段來產生子代架構，突變操作子 (mutation operator) 則根據低的突變率增加子代的多樣性。經歷過上述兩個步驟後，會將親代與子代都放入評價函數 (evaluation function) 來計算均方根誤差 (root mean square error ; RMSE)。由於超級網路先前已經被訓練好了，整個評價流程 (evaluation process) 不需要針對不同的架構從頭開始訓練，而是參考超級網路的共享權重。ASAGA 也設置了溫度機制來加速搜尋流程和避免區域最佳解 (local optimum)，所使用的溫度參數會如同傳統的模擬退火週期性地減少直到搜尋流程結束，Fig. 3 為 ASAGA 的流程示意圖。

Algorithm 1 ASAGA

Input: Supernet S

Output: Best Architecture B

- 1: Initialize: Generation Size G , Initial Temperature T , Final Temperature T_{min} , Annealing Ratio ψ , Architecture Population P , Offspring in genetic algorithm O .
- 2: $P \leftarrow random(S)$
- 3: $B \leftarrow P_{best}$
- 4: **while** $G \neq 0$ AND $T > T_{min}$ **do**
- 5: **for** $n=0$ to $\frac{N}{2}$ **do**
- 6: Select Two Architecture: $p \leftarrow random(P)$
- 7: $O \leftarrow crossover(p)$


```

8:       $O \leftarrow \text{mutation}(O)$ 
9:      for  $i=1$  to 2 do
10:          $\delta \leftarrow \text{evaluate}(p[i]) - \text{evaluate}(O[i])$ 
11:         if  $\delta > 0$  then
12:             $p[i] \leftarrow O[i]$ 
13:         else if  $\text{rand} < e^{\frac{\delta}{T}}$  then
14:             $p[i] \leftarrow O[i]$ 
15:         end if
16:      end for
17:       $P \leftarrow \text{update}(P, p)$ 
18:  end for
19:  if  $\text{evaluate}(B) > \text{evaluate}(P_{best})$  then
20:      $B \leftarrow P_{best}$ 
21:  end if
22:   $T \leftarrow T * \psi$ 
23:   $G \leftarrow G - 1$ 
24: end while
25: return  $B$ 

```

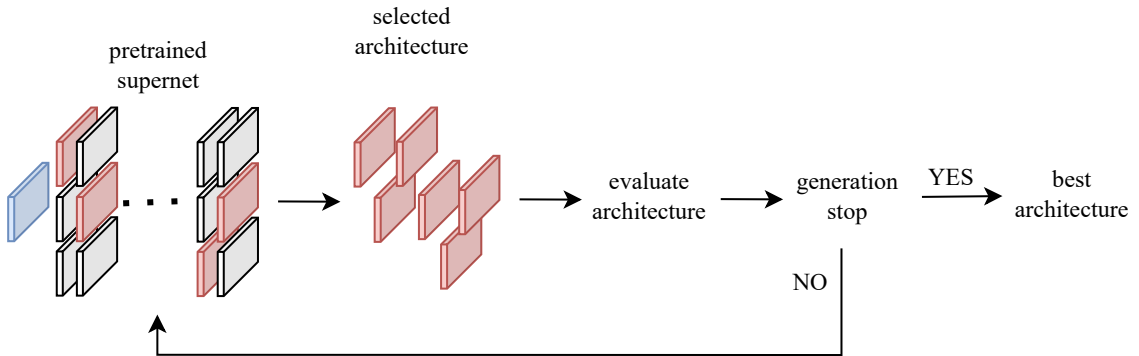


Fig. 3: ASAGA 的運作流程。在每一世代中 ASAGA 會從訓練完成的超級網路隨機選擇架構進行評估，並在最後一個世代結束時搜尋出表現最佳的模型架構。

4.3 興趣指數

除了車流量和站點車輛數目，SAGAON 也採用了鄰近站點的興趣指數作為輸入。我們使用 Google API 的地圖資訊，以站點為中心搜尋方圓 200 公尺內的興趣地點 (interesting spot)，利用笛卡爾座標系統 (Cartesian coordinate system) 將興趣地點的經緯度轉換成對應的座標，藉由興趣地點所構成的城市區域視為一個平行四邊形 (parallelogram) 並定義出 2 種構成該區域的向量，將對應的座標點放入劃分為同樣大小的 $n \times n$ 的網格 (grid)。我們將所有的興趣地點分為 10 類，分別為食物 (food)、購物中心 (mall)、商業 (business)、醫院 (hospital)、大眾運輸 (transportation)、政府組織 (government)、公園 (park)、公寓 (apartment)、娛樂 (entertainment) 和運動 (sport)。每個站點有 10 張不同種類的興趣點地圖 (POI map)，表示成 $M_i \in \mathbb{R}_i^{10 \times n \times n}$ ，每張地圖代表特定的種類 a 。 $\mu_{i,x}^a$ 為站點 i 、種類 a 在興趣點地圖座標 x 上的評價總數 (rating amount) 總和值。因應不同座標上評價總數值可能存在極大的差距，我們使用極值正規化 (min-max normalization) 來處理評價總數：

$$\mu_{i,x}^{a'} = \frac{\mu_{i,x}^a - \mu_{min}}{\mu_{max} - \mu_{min}} \in [0, 1]. \quad (2)$$

評價總數經過正規化之後會和 ρ 進行運算，站點 i 、種類 a 在興趣點地圖座標 x 的興趣指數公式為：

$$M_{i,x}^a = \sum_{q=1}^{\gamma} (\mu_{i,x}^{a,q'} \times \rho_{i,x}^{a,q}). \quad (3)$$

4.4 最佳化模型超參數

超參數是深度學習一大重要設置，可大幅影響最終的訓練結果。然而我們無法事先得知與當前訓練資料最匹配的超參數組合，因此我們使用 Algorithm 1 的變形，將搜尋的目標從神經網路架構調整為超參數組合，配合二進位編碼 (binary encoding) 來轉換各項不同的參數，方便於實作交配與突變運算子。Fig. 4、Fig. 5 展示了交配運算子與突變運算子的實際流程，類基因演算法中兩者分別代表鄰近 (local) 和全域 (global) 搜尋，嘗試在有限的資源條件下找出最佳的適應解。

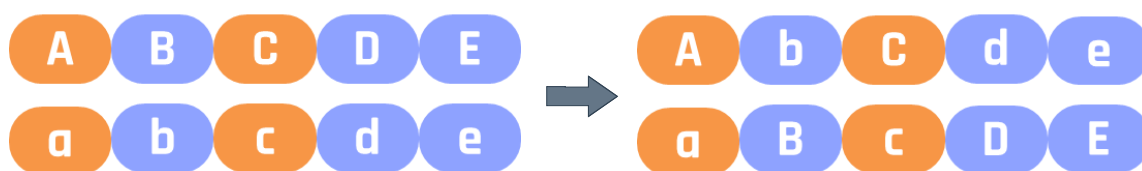


Fig. 4: 交配運算子示例。本專案使用多點 (plural-points) 交配運算，在每次交配時隨機選出多個交配點 (crossover point) 來進行基因交換。在此範例中被選定的交配點用藍色區塊來表示。



Fig. 5: 突變運算子示例。此專案採用多點突變，每次進行突變操作時隨機選出多個突變點進行布林 (Boolean) 反向運算。在此範例中被選定的突變點用淺綠色來示意。

Fig. 6 是自適應模擬退火基因演算法作用於尋找最佳超參數的完整流程。在一開始該演算法會隨機產生一個新的群集以進行交

配與突變的操作，並在評價後用得到的適應度進行模擬退火篩選 (SA-selection)，增加搜尋空間的彈性，避免落入區域最佳解之窠臼。

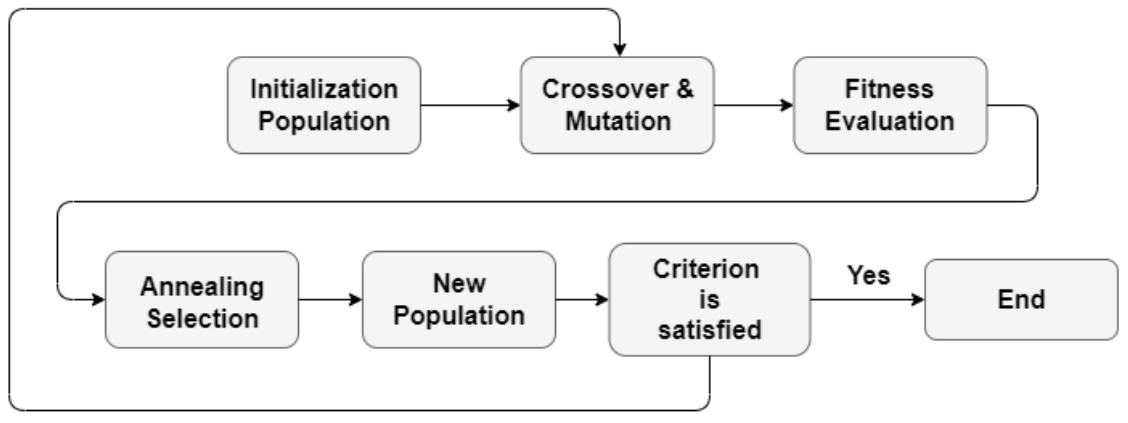


Fig. 6: 自適應模擬退火基因演算法尋找最佳超參數的完整流程。

5 實驗

5.1 環境設置

所有的實驗皆在 Intel Core i5-8400 CPU (2.80GHz)、32 GB DDR4 RAM、NVIDIA RTX-2060 GPU with 6 GB VRAM 的單一臺電腦上執行，作業系統為 Windows 10，使用 Python 3.8.13、Tensorflow-GPU 2.5.0 和 Keras 2.4.3 來撰寫程式。資料集使用的是 New York Citi Bike 於 2020 年 7 月 1 日至 2020 年 8 月 29 日的公開資料，每筆紀錄皆含有開始/結束站點、借車/還車的時間。資料集被分為 2 個部分，2020 年 7 月 1 日到 2020 年 8 月 9 日這 40 天被當作訓練用資料集，剩下的 20 天資料 (2020 年 8 月 10 日到 2020 年 8 月 29 日) 則當作測試用。我們藉由總流量 ($| \text{流入量} | + | \text{流出量} |$) 篩選出前 100 個流量最大的站點作為預測目標，關係密切的站點會被劃分到同一個群組，作為車流量特徵的計算依據。我們還添加了額外特徵如天氣和興趣指數來增加準確率。天氣資訊使用公開網站 (<https://www.wunderground.com/>) 於 2020 年 7 月 1 日到 2020 年 8 月 29 日的觀測資料，每筆紀錄包含了紐約的溫度、降雨、濕度等資訊。我們使用極值正規化來處理這些特徵值並在預測後進行逆正規化 (denormalize)。

5.2 神經網路架構搜尋前處理

5.2.1 再平衡對於公共自行車系統的影響

由於真實資料集中存在的雜訊和資訊缺失，前處理方法也會大幅影響預測的準確率。在實作 New York Citi Bike 資料集前處理時所遇到最大的困難便是站點車輛資料的缺失，紐約市政府並沒有公開各站點的歷史車輛數目，且在進行車輛再平衡 (rebalancing) 時並不會有操作紀錄，若是直接從租借紀錄還原車輛數目會存在落差，從而導致預測失準。為了解決這個問題，我們發現車輛紀錄的連貫性，若第 n 筆紀錄的結束站點與第 $n + 1$ 筆紀錄的起始站點不相同，便能推測這台車輛有經過工作人員再平衡，我們也稱這種情況為幽靈騎者 (ghost rider)。藉由借閱紀錄和再平衡的逆推，我們便能盡量還原真實的站點車輛數目。

5.2.2 車流量

就如先前提到，與 STDN 所使用的前處理方式不同，我們認為使用地理位置建構車流量特徵並不足以精準體現站點間的關係，舉例來說，若有 2 個流量來往密切的站點距離過遠，使用地理位置劃分便有可能忽略這 2 個站點間的相互關係。以站點為基準的前處理方法就是為了解決這項問題而提出的發想，我們首先篩選出了 100 個總流量最大的站點並把它們稱為主要車流站點 (major flow station)，接下來針對每一個目標站點尋找出在特定時間區段內與其擁有密切流量的站點，彙整為該目標的站點群組。為了找出各站點對應的群

組，我們觀察主要車流站點之間的車流量，並將超過最大流量 50% 的站點歸類到預測目標的站點群組，最後以目標站點為中心，找出群組裡離該站點最遠的站點，以它為比例尺進行縮放，並利用笛卡爾座標系統進行轉換，定義出囊括了群組中所有站點的平行四邊形與 2 種構成該區域的向量，把所有站點的車流量特徵標註到 $n \times n$ 的網格裡。

5.2.3 站點車輛數目

利用前述的車流量特徵和再平衡方法，便能夠還原站點車輛數目特徵。我們將資料以每 30 分鐘進行劃分，利用站點間的流量，估算出在時間區段的起始和結束當下目標站點中的自行車數量。如先前所說，由於沒有實際的站點車輛數，上述流程推估出來的為相對的車輛浮動值。為避免在逆推時造成腳踏車數量負值，我們會對預測值進行檢查，若為負值則會進行平移以確保預測值合理。

5.2.4 參數設定

在我們的實驗中，批次大小 (batch size) 設定為 64，學習率 (learning rate) 則是 0.001。我們在訓練已經找到的神經網路架構時搭配了使用了停止 (early stop) 機制。我們設定自適應模擬退火基因演算法的世代 (generation) 數目為 10、族群則為 20。初始的溫度為 1，搭配上 0.9 的冷卻率 (cooling ratio)。在評價結果時我們會過濾掉站點車輛數目值時小於 15 的部分。

5.3 超參數優化前處理

5.3.1 編碼設計

在 STDN 之中共有 5 種使用者可自行更動之參數，分別為注意力長短期記憶模型數量 (att_lstm_num)、長效型長短期記憶模型數量 (long_term_lstm_num)、短效型長短期記憶模型數量 (short_term_lstm_num)、鄰近區域大小 (nbhd_size) 與卷積神經網路鄰近區域大小 (cnn_nbhd_size)。這些可調整之參數都是十進位的整數，為使其能夠進行交配和突變此類基因演算法所需操作，我們對這些參數採用二進位的編碼方式。Fig. 7 是由 STDN 超參數所構成的染色體示意圖。

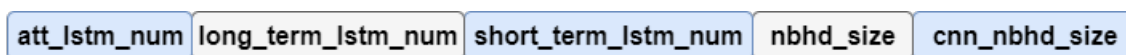


Fig. 7: STDN 參數轉換成適用於該框架之染色體之預覽圖。

5.3.2 參數設定

我們設定自適應模擬退火基因演算法的世代數目為 5、族群大小則為 6。交配與突變的機率分別為 0.8 與 0.2。初始的溫度為 1.5，搭配上 0.8 的冷卻率 (cooling ratio)。每個區塊的長度根據事前規定的參數範圍來變動，Table 2 為該專案中在所有超參數的範圍規定。由於 STDN 的原始設計，注意力長短期記憶模型數量、長效型長短期記憶模型數量和短效型長短期記憶模型數量的數值必須大於 1，且長效型長短期記憶模型數量必須為奇數。我們使用事先定義好的上下限來組成左閉右開的半開區間 (half-open interval)。參數的上限值依實驗

設備可進行適量地調升，然而大部分深度學習模型訓練所需時間與資源與參數大小呈正相關，為證明該策略有效性，我們期待能夠以一組數值更小的超參數來獲得同等於或優於原論文成果，因此在此實驗中各項超參數的上限即為原論文所使用的預設參數。

Table 2: 超參數範圍

參數名稱	下限	上限
注意力長短期記憶模型數量	2	4
長效型長短期記憶模型數量	3	8
短效型長短期記憶模型數量	2	8
鄰近區域大小	0	4
卷積神經網路鄰近區域大小	0	4

5.4 實驗結果

5.4.1 神經網路架構搜尋成果

Table 3和 Fig. 8展示了 SAGAON 對比其他基線實驗 (baseline) 在紐約市公共自行車資料集的預測結果，每種方法都執行了 5 次並紀錄平均值和標準差。SAGAON 勝過了所有的基線實驗，均方根誤差跟平均絕對百分比誤差 (mean absolute percentage error ; MAPE) 在測試資料集上皆為最低。HA 和 ARIMA 這類的傳統時序預測方法準確率非常糟糕，因為它們只依靠過去的歷史紀錄來預測，忽略了空間特徵。SAGAON 也勝過了長短期記憶模型、卷積長短期記憶模型

和 STDN 這類深度學習方法，潛在原因可能是 SAGAON 對於空間與時間特徵的處理更為妥當。長短期記憶模型只關心過去連續的時序資料，忽略了空間資料；卷積長短期記憶模型使用卷積層來捕捉空間特徵，並保留了長短期記憶模型的優勢。STDN 使用空間-時間相混的資料來進行訓練，用以捕捉了動態空間相似性 (dynamic spatial similarity) 和週期時間遞移性 (periodic temporal shifting)，這也是它的結果能夠勝過其他模型的決定性因素。以 STDN 為基礎，我們提出的 SAGAON 增添了 one-shot NAS 的超級網路，藉由選擇不同的選擇模塊來增加訓練的準確率，同時因為透過超啟發式演算法 ASAGA 作為搜尋策略，SAGAON 在搜尋架構的過程中可以減少許多時間和計算資源。

Table 3: 基線實驗和 SAGAON 之間的比較

	Start		End	
Method	RMSE ↓	MAPE ↓	RMSE ↓	MAPE ↓
ARIMA	11.43±0.00	73.89±0.00	11.78±0.00	75.99±0.00
HA	8.56±0.00	67.24±0.00	8.57±0.00	67.51±0.00
LSTM	2.97±0.73	17.66±0.29	4.13±0.12	25.59±0.74
Conv-LSTM	2.13±0.31	11.11±1.23	3.36±0.26	19.88±0.44
STDN	2.07±0.23	10.83±0.80	3.33±0.16	20.23±0.90
SAGAON	2.07±0.03	10.81±0.12	3.31±0.06	19.10±0.17

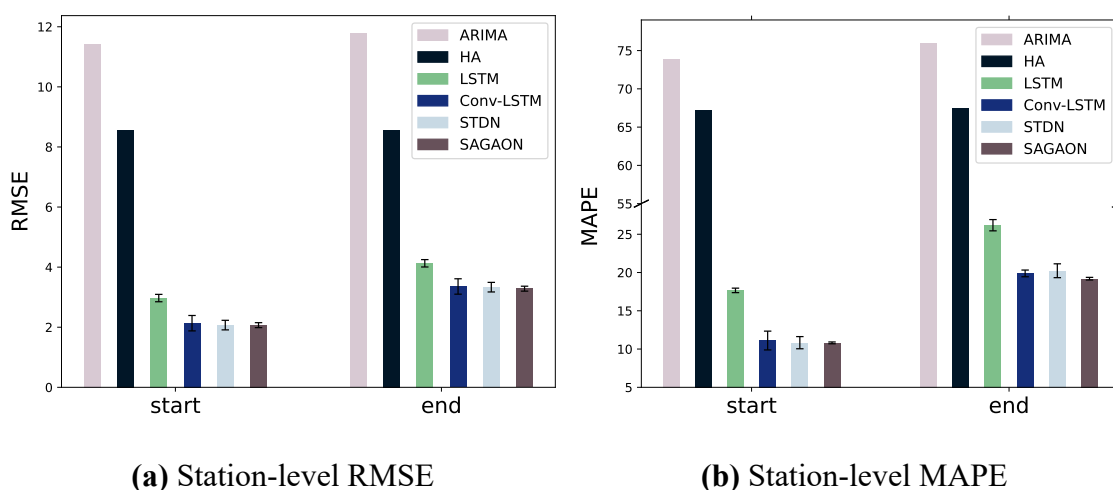


Fig. 8: SAGAON 對比不同的基線實驗模型

5.4.2 超參數優化成果

Table 4 是論文預設實驗與套用了超啟發演算法參數搜尋策略的對比。每種方法都執行了 3 次並紀錄平均值和標準差。傳統的基因演算法及我們提出了自適應模擬退火基因演算法皆得到比預設參數更好的實驗結果。SAGA 在預測租借車輛的部分擁有較亮眼的表現，GA 則是在歸還車輛的部分有突出的效果。SAGA 與 GA 所選定的超參數序列皆小於預設參數，證明了在超啟發演算法的幫助下我們能夠以更少的運算資源取得更好的表現。此優化得益於超啟發演算法的泛用性，無需使用者先行理解該框架的背景知識即可運用。

5.5 使用者介面設計

為了視覺化我們的實驗結果，我們架設了一個簡單的網站來呈現模型在每個不同時間區間對每個站點預測的結果，並且以顏色來

Table 4: 預設參數和採用了超啟發演算法搜尋策略之間的比較

	Start		End	
Method	RMSE ↓	MAPE ↓	RMSE ↓	MAPE ↓
Default	8.85 ± 0.11	21.84 ± 0.36	8.15 ± 0.15	20.87 ± 0.39
Genetic Algorithm	8.83 ± 0.06	21.60 ± 0.21	8.14 ± 0.03	20.42 ± 0.18
Simulated Annealing Genetic Algorithm	8.81 ± 0.04	21.63 ± 0.16	8.15 ± 0.07	20.66 ± 0.04

區分預測結果之好壞。我們運用開源的框架及模組來架構網頁基礎，搭配應用程式介面 (application programming interface ; API) 來快速構建出所需的視覺呈現效果。我們使用了以下的開源工具來完成這項工作：

- 框架 → Django：

Django 是一個以 Python 所建立的模組模板可視化 (model-template views ; MTV) 網路框架。

- 非同步請求 (asynchronous request) → Ajax、Asynchronous JavaScript 與 XML：

Ajax 和 Asynchronous JavaScript 是一個能讓網站能在不用刷新頁面的情況下自動從資料庫抓取資料的一個請求方法 (request method)。

- Javascript 第三方程式庫 → chart.js：

我們引用了 chart.js 這個函式庫輔助網頁上視覺化作圖的部分。

- 資料庫 → MySQL：

我們建立了一個專用的 MySQL 資料庫用以存各站點的所有腳踏車數量，包含了真實數量 (ground truth) 及模型預測數量 (predict value)。

- 應用程式介面 → Google Maps JavaScript API:

Google Maps JavaScript API 用來呈現美國紐約市的地圖，並以客製化標記 (custom marker) 來代表各個目標站點。

Fig. 9是成果頁面的整體架構。前後端的部分皆是使用 Django 預設的網路框架。在原先計畫中有考慮使用 React 或是 Angular 等更細緻化的前端框架，但考慮到時間成本因此最後並沒有採用。

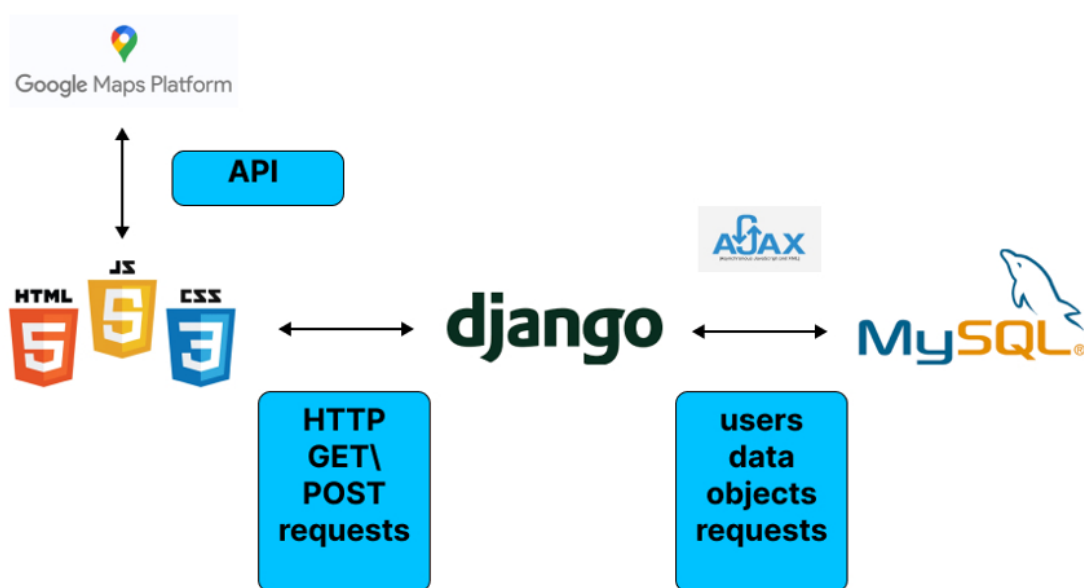


Fig. 9: 網頁整體框架。

為了負荷後端系統龐大資料量的輸入輸出，我們採用了非同步

系統的架構。非同步讓我們可以在不主動更新頁面的情況下，利用 Ajax 來更新網頁內容。舉例來說，Instagram 與 Facebook 等大型社交媒體能夠在不重新整理頁面的前提下獲得新的內容，這就是 Ajax 的基本應用。Fig. 10 簡單示意了同步與非同步請求的差別。可以看到傳統的同步請求必須在一次請求之後必須馬上回覆一次，這在處理大量資料時會造成等待時間過久，非同步請求則能有效解決掉這方面的困擾。

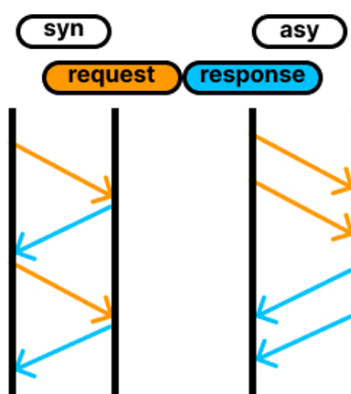


Fig. 10: 同步請求與非同步請求的差別。

6 結論

在這項專題研究中，我們提出了模擬退火基因演算法 one-shot 神經網路搜尋 (SAGAON)，一個針對公共自行車系統站點車流進行預測的神經網路搜尋框架。此框架藉由自動找尋出擁有高準確率的模型架構，改善了空間-時間相依性高的深度學習模型對於真實世界交通流問題的預測能力。根據資料集的測試，SAGAON 即使與其它最先進的方法相比依然有相當高的競爭力。目前所使用的訓練資料為紐約市 Citi Bike 的歷史紀錄，我們在未來打算將該框架實際應用於高雄市的 YouBike 系統，嘗試將此系統與使用者介面應用在台灣 YouBike2.0，這個使用者介面不僅能讓嘗試解決調度問題的系統開發人員有一個工具來了解各個時段、區域、天氣條件下各站點之狀況，也可以讓幕後人員能更快掌握整個系統當前情況。幫助相關的民間業者和政府組織改善公共自行車系統的使用狀況和效率。再者，我們將致力於改善該框架的訓練成本如訓練所需的計算資源與時間，如此一來便能讓 SAGAON 在更多資源有限的終端上運行。

7 評審意見與回覆

Q: 在資料呈現頁面可以發現某些特定站點在大部分時間段的準確率都不是很理想，請問有針對這部分做甚麼處理嗎？

A: 預計未來觀察不準站點的特徵來確立改善方向。另外目前的訓練資料只集中在兩個月，會再進行不同月份資料的訓練後進行觀察。

Q: 可以發現所使用的資料時段並不是當下的，是否可以把框架應到即時處理？

A: 目前我們所使用的原始資料 (raw data) 在釋出上有延遲，導致比較難以應用在即時處理上。或許用其他的方法如主動性領域適應 (active domain adaptation) 能夠解決跟問題。

Q: 是否有針對不同特徵值的處理及實驗結果？哪一些特徵值對結果的影響比較重要？

A: 在經過有無興趣指數和天氣特徵的實驗後，發現興趣指數對於準確度的影響較天氣明顯，不過某些條件下加入外部特徵會造成負面影響。

Q: 論文中提到該方法可以減少搜尋所需的時間與計算資源，在這方面是否有相關的論述與證明？

A: 經過測試 220 組不同模型架構的估算與實驗後，不使用 one-shot 方法的搜尋時間約為 146 小時，SAGAON 大概為 17 小時，

由此說明 SAGAON 可以在更少的時間內完成相同複雜的搜尋流程。

Q: 可以詳述如何把站點座標轉換成網格座標的嗎？

A: 在4.3與5.2.2提到了如何把興趣地點與目標站點的經緯度轉換成對應的網格座標。使用平行四邊形勾畫目標區域並找出 2 種構成該區域的對應向量，劃分出需要的網格大小以此完成座標化。

Q: 實驗數據是否能補上更詳細的參數解釋與說明？

A: 5.2.4提供了關於我們所使用的詳細參數設定。

Q: Algorithm 1缺少了某些參數 (像是 δ) 的說明，是否可以稍微解釋？

A: δ 指的是父代與子代之間適應度 (fitness) 的相減值，用以判斷子代是否比親代有更好的適應度。其餘參數則在 Algorithm 1 第 1 行有標示。

Q: 為什麼你們的模型是先壓縮空間特徵而不是時間特徵？請問這樣做的用意在於？

A: 使用 CNN 能夠優先過濾出重要的空間特徵，再把這些經過篩濾的資料送入 LSTM 進行時間相關的預測，通常這樣做的效果會比較好。未來可以嘗試改變模型來比較兩者不同之處。

8 致謝和聲明

這項研究受到中華民國國家科學及技術委員會的協助，計畫編號 NSTC 111-2813-C-110-033-E 和 NSTC 111-2222-E-110-006-MY3。關於前處理和模型的詳細程式碼，我們分別在 (https://github.com/YunYe0121/CitiBike_preprocessing) 和 (<https://github.com/B083040012/SAGAON>) 有更詳盡的說明。該專案之研究結果已整理齊全並已被 IEEE Applied Sensing Conference [33] 所接受，預計 2023 年 1 月於該研討會口頭發表。

9 參考文獻

- [1] D. Chemla, F. Meunier, and R. W. Calvo, “Bike sharing systems: Solving the static rebalancing problem,” *Discrete Optimization*, vol. 10, no. 2, pp. 120–146, 2013.
- [2] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.
- [3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [5] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS’11, p. 2546–2554, 2011.
- [6] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised learning,” *International journal of computer science*, vol. 1, no. 2, pp. 111–117, 2006.

- [7] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, “Data preprocessing and intelligent data analysis,” *Intelligent data analysis*, vol. 1, no. 1, pp. 3–23, 1997.
- [8] J. Waring, C. Lindvall, and R. Umeton, “Automated machine learning: Review of the state-of-the-art and opportunities for healthcare,” *Artificial Intelligence in Medicine*, vol. 104, p. 101822, 2020.
- [9] T. Elsken, J. H. Metzen, and F. Hutter, “Neural architecture search: A survey,” *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.
- [10] H. Zhao, L. Wei, and Q. Yao, “Simplifying architecture search for graph neural network,” *arXiv preprint arXiv:2008.11652*, 2020.
- [11] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, “Understanding and simplifying one-shot architecture search,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80, pp. 550–559. PMLR, 2018.
- [12] S. Narmadha and V. Vijayakumar, “Spatio-temporal vehicle traffic flow prediction using multivariate cnn and lstm model,” *Materials Today: Proceedings*, 2021.
- [13] Y. Liu, H. Zheng, X. Feng, and Z. Chen, “Short-term traffic flow prediction with conv-lstm,” in *2017 9th International Conference on*

Wireless Communications and Signal Processing (WCSP), pp. 1–6, 2017.

- [14] A.-S. Mihaita, H. Li, Z. He, and M.-A. Rizoïu, “Motorway traffic flow prediction using advanced deep learning,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1683–1690, 2019.
- [15] C. Park and S. Y. Sohn, “An optimization approach for the placement of bicycle-sharing stations to reduce short car trips: An application to the city of seoul,” *Transportation Research Part A: Policy and Practice*, vol. 105, pp. 154–166, 2017.
- [16] E. H.-C. Lu and Z.-Q. Lin, “Rental prediction in bicycle-sharing system using recurrent neural network,” *IEEE Access*, vol. 8, pp. 92 262–92 274, 2020.
- [17] Y. Zhou and Y. Huang, “Context aware flow prediction of bike sharing systems,” in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2393–2402, 2018.
- [18] Y. Yang, A. Heppenstall, A. Turner, and A. Comber, “Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems,” *Computers, Environment and Urban Systems*, vol. 83, p. 101521, 2020.
- [19] Y. Lv, Y. Duan, W. Kang, Z. X. Li, and F. Wang, “Traffic flow pre-

- diction with big data: A deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 865–873, 2015.
- [20] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” *Advances in neural information processing systems*, vol. 28, 2015.
- [21] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, “Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, pp. 5668–5675, 2019.
- [22] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” *arXiv preprint arXiv:1611.01578*, 2016.
- [23] B. Wang, Y. Sun, B. Xue, and M. Zhang, “Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification,” in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2018.
- [24] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving deep convolutional neural networks for image classification,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, 2020.
- [25] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Smash: one-shot

model architecture search through hypernetworks,” *arXiv preprint arXiv:1708.05344*, 2017.

- [26] F. Pedregosa, “Hyperparameter optimization with approximate gradient,” in *Proceedings of the International conference on machine learning*, pp. 737–746, 2016.
- [27] H. Liu, K. Simonyan, and Y. Yang, “DARTS: Differentiable architecture search,” *arXiv preprint arXiv:1806.09055*, 2018.
- [28] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, “Single path one-shot neural architecture search with uniform sampling,” in *Proceedings of the European Conference on Computer Vision*, pp. 544–560, 2020.
- [29] T. Bäck and H.-P. Schwefel, “An overview of evolutionary algorithms for parameter optimization,” *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.
- [30] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, “Regularized evolution for image classifier architecture search,” in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, pp. 4780–4789, 2019.
- [31] Y. Li, Y. Zheng, H. Zhang, and L. Chen, “Traffic prediction in a bike-sharing system,” in *Proceedings of the 23rd SIGSPATIAL Interna-*

tional Conference on Advances in Geographic Information Systems, pp. 1–10, 2015.

- [32] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [33] B.-H. Chen, Y.-Y. Cai, C.-Y. Huang, and C.-W. Tsai, “An effective evolutionary neural architecture search for bike-sharing system demand prediction,” in *Proceedings of the IEEE Applied Sensing Conference*, 2023.